



# *Technology Audit*

## WebObjects

Web Development Series

Published December 1996

*This is an independent analysis of WebObjects v3.0 and is the result of proprietary research by **Butler** Group. The Technology Audit Series includes software tools, database, middleware and software architectures, further information can be obtained from the subscription manager on +44 (0)1482 642700.*

*Comments on any **Butler** Group publications should be forwarded in writing to the Quality Manager.*



## Summary

**The Internet has provided the ideal vehicle for low cost electronic commerce, and Butler Group believe that it is this above all else driving the shift towards a real-time economy**

The transition from the Industrial Age to the Information Age has been well documented. Whereas mass markets meant that mass production and improved efficiency were once the optimum business model, increasingly discerning consumers have forced a shift towards shrinking markets, mass customisation and high differentiation. Organisations have had to cut costs whilst at the same time, improving the quality of goods and services and increasing the speed of delivery. Butler Group believe that electronic commerce is the key to success through its inherent ability to reduce costs, lower time to market, consumer response times, and improve the quality of services. The Internet has provided the ideal vehicle for low cost electronic commerce, and Butler Group believe that it is this above all else driving the shift towards a real-time economy.

Although Internet technology has been around for decades, it was not until the advent of the Web (the graphical face of the Internet), that interest really grew. The first generation of applications for the Web were very basic, providing static on-line marketing information with little or no user interaction. However, because this is now the norm rather than the exception, it is unlikely to provide competitive advantage. Creation of a 'cool' Web site requires increasingly sophisticated techniques and has become something of an administrative nightmare for those seeking to gain competitive edge in this way. However, the Web can be utilised in a far more strategic manner, as the latest computing platform for business applications. Whereas the first Web applications were simple forms based affairs, the technology has developed to an extent that it is feasible to deploy fully interactive applications on the Web.

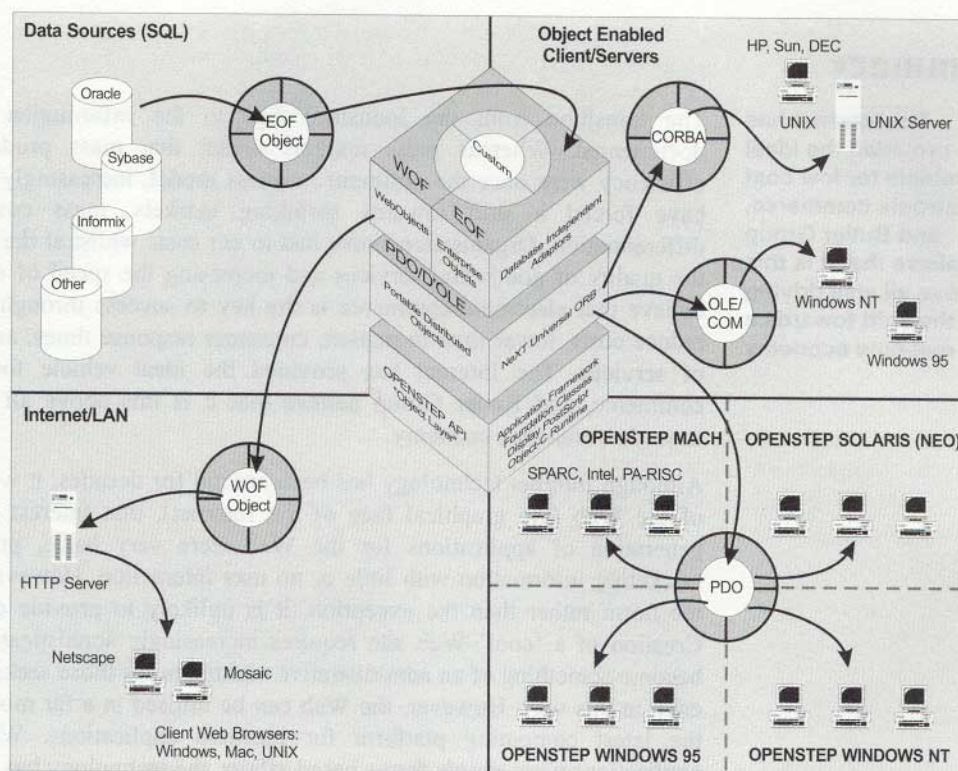
Until recently, one of the major inhibitors of development for the Web was the need for the programmer to have an understanding of a variety of languages and tools. Building the user interface required a certain amount of HTML knowledge and familiarity with the CGI (Common Gateway Interface) was necessary to support input/output of data. Application logic could be written in any standard programming language, but familiarity with SQL was also necessary to provide database access. Even if the CGI script was written in the same language as the application logic, there was a significant learning curve and time lapse before development even began. Many applications are still custom built in this way, but the scalability and maintainability of such applications does fall into question.

**WebObjects provides a fully OO development environment based on already established products, making it more of an evolutionary rather than revolutionary product**

Over the last twelve months, many of the established client/server application development vendors have offered some form of Web development tool and NeXT Software are no exception. NeXT have been firm advocates of OO technology for many years, and are best described as leaders rather than followers in this area. WebObjects provides a fully OO development environment based on already established products, making it more of an evolutionary rather than revolutionary product. NeXT see the Web as the latest in a long line of computing platforms; a natural development of OO technology.

NeXT believe that the underlying problem with application development is that of keeping database applications synchronised with the frequent policy and procedure changes, necessary to achieve a realistic degree of responsiveness. By fusing the data and business methods into an object, business applications are easier to build, deploy and maintain. The method taken by NeXT's Enterprise Objects (EO) Framework is to generate objects automatically at run-time from the developer's definitions. The EO Framework means that the only coding needing maintenance is that defining the business methods. In short, this Framework is able to link relational data with OO development and integrate them into any OPENSTEP (an open API based on NEXTSTEP OO technology) compliant development environment.





Whereas the EO Framework takes care of back end data access, irrespective of deployment platform, WebObjects enables the developer to be insulated from the complexities of building Web applications; HTML is automatically created on the fly, based on user response. And, because developers need only concentrate on implementing the business logic, rather than presentation and data access aspects of Web applications, the development cycle is reduced significantly.

WebObjects unsurprisingly adheres to the request/response model of the Web, and applications are based on pages. These applications contain three major components: HTML templates, declaration files and scripts. The template encapsulates the HTML elements that define the layout and structure of the page, whilst WEBOBJECT mark-up elements may be included to be replaced with dynamically produced HTML upon execution of the program. The use of proprietary HTML tags is seen to be controversial by some, at a time when standards are emerging and redefining themselves at an unprecedented rate. However, these tags are internal to WebObjects and not visible to the browser. Declaration files determine the components that dynamically generate the HTML to substitute for the WEBOBJECT mark-up elements, and finally, the script files comprise of business logic specific to the application (generally defining actions to a response to user requests).

Unlike a number of other tools, WebObjects offers an open architecture that allows it to function with a wide range of third party products. It is browser independent, providing extended support for enhanced features such as HTML 3.0, proprietary HTML extensions, and Java applets, in addition to SSL and S-HTTP security links. It is not dependent on any particular Web server, the only set criteria being that it supports CGI. If optimum performance is required, then the application can be migrated to take advantage of server specific APIs - without the need for additional coding.



**WebObjects application will reside on the same machine as the HTTP server, in a small configuration, as loading increases, it can be moved to a dedicated application servers**

The issue of scalability is handled in a number of ways. If heavy loading begins to effect performance, it is possible to create additional distributed Web servers. Whilst the application partitioning also allows database intensive processes to be moved onto high performance servers for maximum efficiency gain. Whilst the WebObjects application will reside on the same machine as the HTTP server, in a small configuration, as loading increases, it can be moved to a dedicated application server(s)

Web applications can be integrated into corporate data irrespective of where it is stored, so protecting investment in existing resources. Applications are able to access multiple data sources, including Oracle, Sybase, Informix, DB/2 and flat files. WebObjects can integrate with windows applications using OLE, and mainframe applications through third party applications.

Previously criticised for its unsophisticated interface handling, most notably the lack of a visual HTML layout tool. WebObjects has, from version 3.0, introduced a much friendlier graphical development environment (code named Tsunami). This goes beyond basic HTML editing, enabling the developer to create the entire application from within a single environment. It allows the Web developer to visually design pages by dragging and dropping objects (developed using a variety of object models), whilst easing the task of connecting them to the business logic and data source. A palette of re-useable components (HTML and application logic objects) is provided along with support for developing your own Java classes.

To summarise, Butler Group believe WebObjects is one of the first industrial strength development tools for building, and more importantly maintaining, large dynamic Web applications that require long-lived user sessions, transaction management and database access. Its main design goal can be summarised in one word: dynamism. Non OO developers will, however, have to overcome a relatively steep learning curve. Having said that, Butler Group believe Web development, much like client/server, is well suited to the OO paradigm.

## Strengths

- Based around established OO technology
- Open architecture facilitates third party integration
- Application logic not tied to the user interface or the database
- Enterprise strength scalability
- Strong state management functionality

## Weaknesses

- Steep learning curve for inexperienced OO developers
- No integrated web server for testing purposes
- Market presence and profile needs to be raised



## Market Position

The IT industry is never slow to take advantage of possible new opportunities and as a result of shrewd marketing, they have managed to create a multi-million Internet related market. Until recently, the Web was used merely as a marketing medium, concerned with the publication of static documents. However, the number of fully interactive business applications deployed on the Web is increasing at an unprecedented rate.

Only a year ago, building Web applications was a lengthy and time consuming process involving specialist programmers. By automating many of the complexities associated with Web development, the new breed of Web enabled application development tools allow simple applications to be built in a matter of hours and days, rather than months and years.

As the industry moves away from traditional Web development techniques, i.e. CGI/Perl hacking, we are witnessing a deluge of new "web-enabled" development tools. It will not be too long before every major application development tool vendor will have added some Web development functionality. Being one of the first, NeXT must look to take advantage of this lead by increasing their market profile and presence sooner rather than later. It is of little use having a technically sound product without a significant degree of market awareness. Although WebObjects will appeal primarily to those already developing in a NeXT environment, NeXT as a company must try to eliminate the image that they are a "niche" outfit, approachable only by true OO gurus. Any developer familiar with the OO paradigm should be able to make the transition to WebObjects relatively easily and quickly. Those with no OO knowledge and with short timescales may, however, be better advised to migrate to the OO paradigm at their own pace using an "object-based" tool.

Butler Group believe WebObjects is a Web development tool that is aimed at the corporate rather than casual Web developer. The application infrastructure differentiates it from the lower end of the market. Examples of applications built using WebObjects can be obtained from NeXT at <http://www.Next.com>.

For creating stand-alone dynamic Web applications, the basic product can be obtained free of charge from the NeXT Web site. WebObjects Pro offers scalability, performance and access to legacy data for \$2,999 per processor, and finally, WebObjects Enterprise provides seamless integration into the corporate infrastructure at \$24,999 per processor, per server.

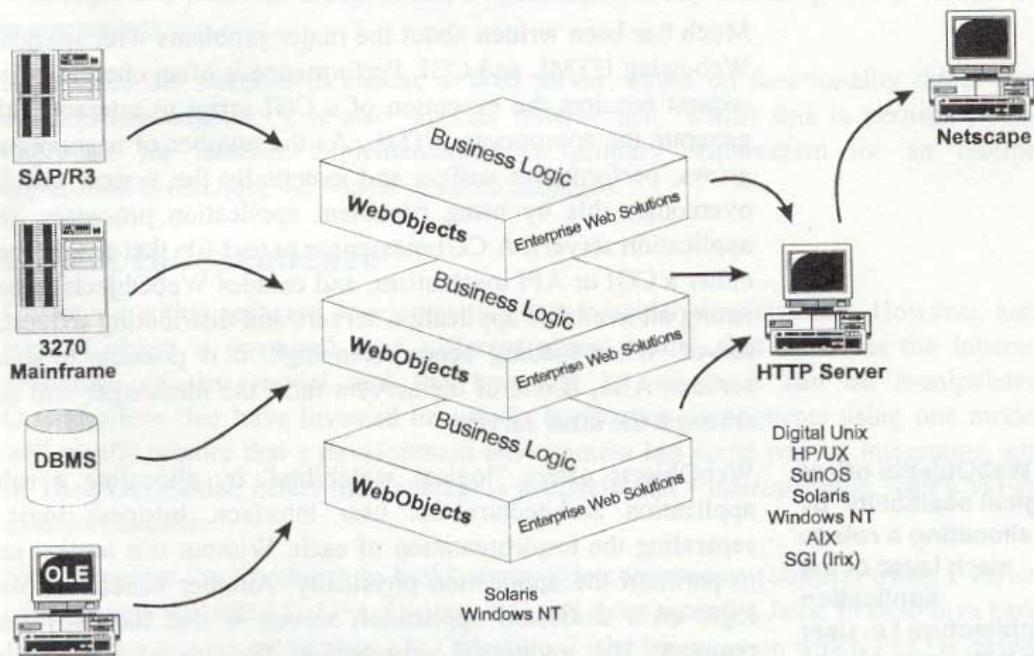
**Butler Group believe WebObjects is a Web development tool that is aimed at the corporate rather than casual Web developer**



## Architecture

WebObjects is based largely on proven NextStep technology including the OpenStep specification, Portable Distributed Objects (PDO) Object Model and the Enterprise Objects (EO) Framework. Unlike the traditional single-tier architectures concerned with CGI scripting or stored database procedures, WebObjects opts for a three-tier approach to building Web applications. It is the EO Framework that enables WebObjects to operate in a multi-tier environment by separating presentation, business logic and data access. The process of combining business logic and HTML is handled automatically, dependent on the user response, as is back-end data access, leaving the developer to concentrate on business logic - where the true added value exists.

NeXT are firm believers in component based software development and utilise a framework-based approach as opposed to traditional procedural methods. Although the underlying Framework can accommodate the requirements of most database applications, developers are also able to modify the Framework by changing the behaviour of objects, adding classes, or substituting Framework objects with their own or third party components. By using pre-built components and/or Java applets, developers can create complex applications using a minimum of scripting. Where scripting is necessary, WebObjects provides its own scripting language (WebScript) and support for VBScript and JavaScript.





## Technical Features

### Platforms and Connectivity

#### Client

- Most Web Browsers
- Windows family
- NeXT's Mach OS

#### Application Server

- Windows NT
- Solaris
- SunOS
- HP-UX
- Digital Unix
- NEXTSTEP

#### Web Servers

- Any with CGI, ISAPI or NSAPI interface

#### Database

- Oracle
- Sybase
- Informix
- DB/2
- Others (through gateways)

### Performance/Scalability

Much has been written about the major problems with application development for the Web using HTML and CGI. Performance is often cited as a problem because each user request requires the execution of a CGI script to interact with the data source and to generate the appropriate HTML. As the number of users requesting application access grows, performance suffers and eventually the system grinds to a halt. WebObjects overcomes this by using persistent application processes, (a dedicated WebObjects application server). A CGI messenger, a text file that sits on the HTTP server, links into either a CGI or API mechanism, and enables WebObjects to perform load balancing by listing all available application servers and distributing requests to the most appropriate server. When loading becomes too high, it is possible to dynamically add additional servers. Also, if one of the servers fails, the messenger will simply reallocate requests amongst the other servers.

WebObjects offers "logical scalability" by allocating a role to each layer of the application architecture i.e. user interface, business logic and data access, and separating the implementation of each. Without this logical scalability it is impossible to partition the application physically. Another benefit of placing all of the business logic on a dedicated application server, is that the HTTP server can service more requests. The traditional approach to Web development necessitated that the Web applications were run on the HTTP server, which meant additional HTTP servers were added to improve performance under strain. It is a far simpler task to add application servers, than to go through the rigmarole of registering additional HTTP servers.

### State Management

HyperText Transport Protocol (HTTP) is a client/server protocol used for information sharing on the Internet; it is the foundation of the World Wide Web (WWW). HTTP is a stateless protocol, and consequently information about past or concurrent operations cannot be stored. This is adequate for publishing documents on the Web - the user simply sends a request for a resource, which is then processed and returned to the browser - the session is then terminated.

WebObjects offers 'logical scalability' by allocating a role to each layer of the application architecture i.e. user interface, business logic and data access, and separating the implementation of each



However, this is of no use for even the most routine business transactions, which require a connection (state) until the process is completed. WebObjects provides a state management system to help the developer manage data more effectively; this is done primarily through the use of hidden variables. State can be maintained at a number of levels, from a global state (where state is maintained for the duration of the application e.g. connection to data sources), to a session state (where state lasts until a session is completed) down to the more short-lived page and transaction variables.

## Integration with legacy applications

WebObjects allows organisations to leverage investment in existing applications and data. Using NeXT's proprietary ORB technology, OLE, DCE and CORBA, WebObjects can integrate with existing client/server applications and utilise application specific information. Integration with mainframe applications is made possible through the use of third party products. In addition, WebObjects provides access to a number of popular databases including Oracle, Sybase, Informix and DB/2, without the need for additional coding; others are available via ODBC. As WebObjects applications are totally independent of data source, protection of the current data infrastructure is guaranteed.

## Utilisation of Existing Technology

It is simple fact that many organisations do not standardise on a particular platform, database, etc. but instead use a plethora of technologies built up over the years. WebObjects is platform independent, a necessity in today's heterogeneous computing environments.

It provides the freedom to choose a Web server, based on functionality rather than development support; it is also browser independent. Whilst this is perhaps not so important for Intranet implementations, a primary requirement for an Internet application is reaching the widest possible audience.

## Object Independence

Object use within software development is now considered mainstream. However, each type of object is governed by a different object model, that describes the inherent properties of that model and the language by which it can be manipulated. Organisations that have invested heavily in application components using one model, will ideally require that a development environment has some way of integrating with it. The PDO model, offers the developers precisely that - interoperation with a variety of object models.

NeXT enables the developer to build server side components (objects) using a variety of languages including C, C++, Objective C and more recently, Java. In fact, Java bares more than a passing resemblance to Objective C (the language on which PDO is based), supporting principles such as dynamic runtime and simplicity. At present, the development of server side Java applications is limited for a number of reasons, these are overcome through the addition of a Java compiler for the NeXT Object Model.

## Security

WebObjects supports both the S-HTTP and SSL security standards, in addition to any other security system on the server, and can also work with existing firewall technology. NeXT offer a sophisticated security mechanism for those planning on Intranet development, through their strategic partnership with Security Dynamics. This is essentially based on smart card technology that ensures the authenticity of user access codes.

NeXT enables the developer to build server side components (objects) using a variety of languages including C, C++, Objective C and more recently, Java



## Vendor Profile

NeXT Computer Inc was founded in 1985 by Steve Jobs. Both he and his senior management team had all previously been involved with Apple, and in worked extensively on the Apple Macintosh. This experience was a solid foundation for the early work of NeXT which was to build and supply its own computers and operating systems.

NEXTSTEP was first released in 1989, running only on NeXT hardware. This proved to be a successful market but needed to grow to meet new industry trends. Thus, in 1993, NEXTSTEP for Intel processors was announced and NeXT ceased building its own hardware.

Later that year, NeXT decided to open the NEXTSTEP operating system by publishing an API called OPENSTEP. Compliant environments provide a distributed, OO application development environment, giving objects and applications using these APIs platform independence across OPENSTEP implementations. Strategic partnerships were later formed with Hewlett-Packard, Sun Microsystems, and Digital Equipment to make both NEXTSTEP and OPENSTEP available to wider range of platforms.

During 1995, NeXT released plans to make OPENSTEP available for Windows 95 and Windows NT. The announcement of WebObjects, an extension of existing NeXT technology for building server based applications for the Web, followed shortly after.

NeXT's mission is to lead the object revolution. It aims to achieve this by capitalising on the widespread move towards object-oriented, client/server and now Internet computing by providing a complete OO development environment.



## Contacts

NeXT Software UK Ltd  
Technology House  
Meadowbank  
Furlong Rd  
Bourne End  
Buckinghamshire  
SL8 5AJ

Tel: +44 (0)1628 535222

Fax: +44 (0)1628 535200



**Butler** Publishing Ltd, Hesslewood Hall, Hessle, Kingston-upon-Hull, HU13 OJF  
Great Britain

Tel: +44 (0)1482 642700 Fax: +44 (0)1482 642691/642694

[HTTP://WWW.BUTLERGROUP.CO.UK](http://www.butlergroup.co.uk)